Supplementary File – Data process & analysis

The raw sequencing data were stored in a Linux server. Bash commands were used to process the data.

1. RNA-seq data quality assessment and trim

RNA sequencing data quality was assessed using FastQC

```
$fastqc sample.fastq.gz
```

Low quality reads and segments were removed using Trimmomatic with leading and tailing threshold: 32, minimum length remains: 50 bps. Here, sample.fastq.gz is the input sequencing file.

```
$trimmomatic SE -threads 10 sample.fastq.gz sample_trimmed_adapter.fastq.gz
ILLUMINACLIP:/trimmomatic-0.39-1/adapters/TruSeq3-SE.fa:2:30:10 LEADING:32
TRAILING:32  MINLEN:50
```

2. Alignment of trimmed reads

The remaining high-quality reads were mapped to the reference genome using bowite2 with default parameters.

```
$ bowtie2 --threads 20  -x refGenome/aeru -U sample_trimmed_adapter.fastq.gz -S
sample_align.sam 1>>log.log 2>> summary.log"
```

3. Mark and remove duplicates

Picard was used to sort the aligned reads, then mark and remove duplicates.

```
$picard SortSam I= sample_align.sam O=sample_sorted.bam SORT_ORDER=queryname

$picard MarkDuplicates I=sample_sorted.bam O=sample_dup_removed.bam
M=sample_dup_metrics.txt REMOVE_DUPLICATES=true
```

4. Mapped read count calculation

Lastly, htseq-count was used to calculate the numbers of reads that mapped to genes. The gene model was provided.

```
$htseq-count -f bam -t CDS sample_dup_removed.bam "208964.12.PATRIC.format.gtf" >
sample_PATRIC.count.txt
```

The differentially expression analysis was then performed using a widely used R package named edgeR. The read count results calculated by htseq-count were fed into this method.

- Lowly expressed genes (median read counts <10) were excluded from further differentially expression analysis.
- The log2 fold change and false discovery rate (FDR) were measured by edgeR. Genes with |log2FC| > 1 & FDR < 0.01 were considered as significant differently expressed genes.

```
setwd("C:/Data/HTSeq count/")
DESample = read.table("sample_PATRIC_1.count.txt", header = F, stringsAsFactors = F)
med.count = apply(mat, 1, median)
DESample_high = DESample[med.count >= 10 ,]
```

```
Library(“edgeR”)

de_edgeR <- function(DEMatrix, n_control, n_treated){ #need to provide the rownames, gene
ids, of the matrix
  group<-c(rep("control",n_control),rep("treated",n_treated))
  cds<-DGEList(DEMatrix,group=group)
  cds<-calcNormFactors(cds)
  cds<-estimateCommonDisp(cds)
  et<-exactTest(cds, pair=c("control","treated"))
  etTable=topTags(et,n=nrow(cds$counts))$table
  return(etTable)
}
```

Identify the differentially expressed genes between treated and control samples. Here we have two samples for each group.

```
de.CB.CP <- de_edgeR(cbind(controls, treated), 2, 2)
sum(abs(de.CB.CP$logFC) > 1 & de.CB.CP$FDR < 0.05)
```